**Laporan Tugas Dasar Pemrograman Kelas A**

Teknik Informatika, FTEIC, ITS, 16 September 2024 (Dosen Pengajar: Diana Purwitasari)

Materi : Percabangan dan Perulangan

***oleh*** Muhammad Fatih Al Fawwaz (NRP 5054241007)

**Topik**:

Soal nomor 10. Membuat kode untuk melakukan enkripsi dan dekripsi menggunakan Caesar Cipher dengan menggunakan pergeseran huruf berdasarkan nilai key.



Gambar 1. Kode nomor 10 untuk enkripsi dan dekripsi menggunakan Caesar Chiper

**Langkah-1**.

T = int(input())

Menerima input integer T yang menunjukkan jumlah kasus uji yang akan diproses.

**Langkah-2**.

for \_ in range(T):

Memulai loop for yang akan berjalan sebanyak T kali. Karakter \_ digunakan sebagai variabel loop karena kita tidak memerlukan nilai indeks itu.

**Langkah-3**.

code, key = map(int, input().split())

Menerima dua input integer yang dipisahkan oleh spasi, yaitu code dan key. code menunjukkan apakah teks akan dienkripsi atau didekripsi: Jika code adalah 1, maka teks akan dienkripsi.

Jika code bukan 1 maka teks akan didekripsi. key adalah nilai kunci untuk proses enkripsi/dekripsi (shift dalam algoritma Caesar cipher).

**Langkah-4**.

kalimat = input()

Menerima input string kalimat yang akan dienkripsi atau didekripsi.

**Langkah-5**.

result = ""

Inisialisasi string kosong result untuk menyimpan hasil enkripsi atau dekripsi.

**Langkah-6**.

for char in kalimat:

Memulai loop untuk setiap karakter char dalam kalimat.

**Langkah-7**.

if char.isalpha():

Mengecek apakah char adalah huruf alfabet (A-Z atau a-z). Jika ya, maka kita akan memproses enkripsi/dekripsi pada karakter tersebut. Jika tidak, karakter akan langsung ditambahkan ke result tanpa diubah.

**Langkah-8**.

if char.isupper():

Mengecek apakah char adalah huruf kapital (A-Z).

Jika code == 1 (enkripsi) maka ord(char) + key - 65: Mendapatkan nilai ASCII karakter char, menambahkan key, dan mengurangi 65 (ASCII 'A') agar perhitungan berbasis indeks 0. % 26: Melakukan operasi modulus 26 agar hasilnya tetap dalam rentang alfabet (A-Z). + 65: Menambahkan kembali 65 untuk mengubahnya menjadi nilai ASCII karakter hasil enkripsi.

chr(...): Mengonversi nilai ASCII kembali ke karakter. Jika code != 1 (dekripsi): Proses yang sama seperti di atas tetapi menggunakan ord(char) - key untuk mengurangi nilai kunci dari karakter tersebut.

**Langkah-9**.

else:

Mengecek apakah char adalah huruf kecil (a-z).

Proses serupa dilakukan seperti di atas, tetapi menggunakan nilai ASCII untuk 'a' (97). Jika code == 1 (enkripsi), hasilnya dihitung dengan (ord(char) + key - 97) % 26 + 97. Jika code != 1 (dekripsi), hasilnya dihitung dengan (ord(char) - key - 97) % 26 + 97.

**Langkah-10**.

else:

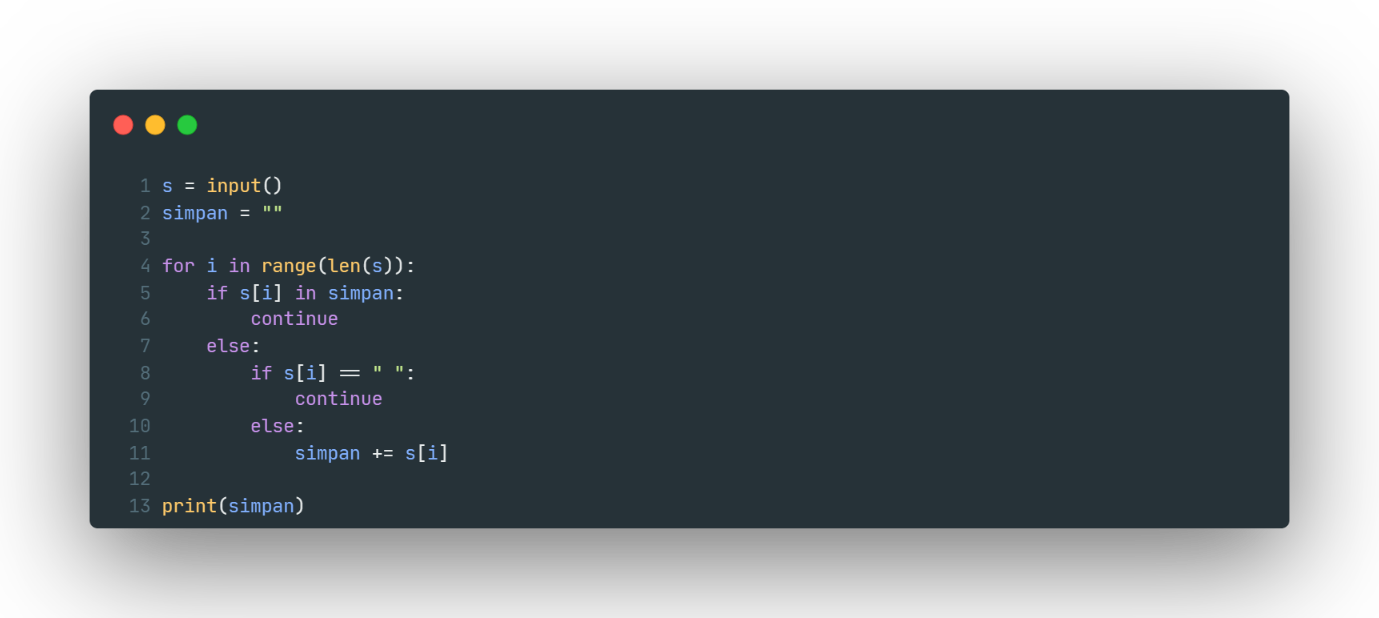
Jika char bukan huruf alfabet, maka langsung ditambahkan ke result tanpa perubahan. Hal ini untuk menangani karakter seperti spasi, tanda baca, atau angka yang tidak perlu dienkripsi/dekripsi.

**Langkah-11**.

print(result)

Mencetak result yang berisi teks hasil enkripsi atau dekripsi setelah loop selesai.

**Topik**:

Soal nomor 11. Membuat kode yang berfungsi untuk menghapus karakter yang berulang dari string, hanya menyimpan karakter pertama kali muncul, dan mengabaikan spasi.

Gambar 2.Gambar untuk soal nomor 11

**Langkah-1**.

s = input()

Menerima input string dengan variabel s dari pengguna.

**Langkah-2**.

simpan = ""

Inisialisasi string kosong simpan yang digunakan untuk menyimpan karakter dari string s.

**Langkah-3**.

for i in range(len(s)):

Memulai loop for yang akan berjalan dari 0 hingga panjang string s. len(s) digunakan untuk mendapatkan panjang string s, dan loop akan memproses setiap karakter dalam string tersebut berdasarkan indeks i.

**Langkah-4**.

if s[i] in simpan:

Mengecek apakah karakter saat ini s[i] sudah ada di dalam string simpan. Jika ya, berarti karakter tersebut sudah pernah ditambahkan ke simpan, jadi kita akan continue, yang artinya melewatkan iterasi saat ini dan melanjutkan ke iterasi berikutnya.

**Langkah-5**.

else:

Jika karakter s[i] belum ada di simpan, maka kita akan memeriksa lebih lanjut.

**Langkah-6**.

if s[i] == " ":

Mengecek apakah karakter saat ini s[i] adalah spasi. Jika ya, maka continue akan dilewatkan karena kita tidak ingin menambahkan spasi ke simpan.

**Langkah-7**.

else:

Jika karakter s[i] bukan spasi dan belum ada di simpan, maka karakter ini akan ditambahkan ke simpan.

**Langkah-8**.

simpan += s[i]

Menambahkan karakter s[i] ke string simpan.

**Langkah-9**.

print(simpan)

Mencetak string simpan yang berisi karakter-karakter unik dari string s (tanpa duplikat dan tanpa spasi).

**Topik**:

Soal BITB (Birthday In The Battlefield). Membuat kode untuk mengeluarkan kue dengan besar sesuai input user

Gambar 3.Kode Untuk soal BITB

**Langkah-1 .**

User memasukkan angka n. Nilai akan menentukan ukuran dari kue. Nilai n harus lebih besar dari 8 untuk menghasilkan pola yang benar.

**Langkah-2.**

Nilai baris dan kolom dihitung berdasarkan n, di mana perhitungan ini digunakan untuk menentukan ukuran total dari bagian tengah pola.

**Langkah-3.**

Baris pertama selalu memiliki 3 baris. Bagian tengah dihitung dengan menggunakan fungsi ceil() untuk membulatkan hasil pembagian (n - 3) agar menjadi lebih besar jika ada nilai desimal. Bagian bawah dihitung menggunakan pembagian // untuk mendapatkan nilai yang lebih kecil atau sama dengan hasil pembagian.

Kolom 1: Bagian pertama hanya memiliki 1 kolom

Kolom 2: Bagian tengah dihitung berdasarkan n, dengan rumus ((n-8)\*2)-1 untuk menyesuaikan ukuran kolom tengah.

kolom3: Bagian bawah dihitung dengan mengurangi kolom dengan 2\*(n-8) untuk menyesuaikan lebarnya.

**Langkah-4.**

Spasi untuk bagian paling atas:

spasi1 untuk Menghitung jarak spasi untuk "roti lapisan pertama" agar simetris di tengah.

spasi2 untuk Menghitung jarak spasi untuk "roti lapisan kedua".

spasi0 adalah spasi awal ditambahkan 1 untuk menempatkan simbol \* di posisi tengah dengan benar.

**Langkah-5.**

Jika n kurang dari atau sama dengan 8, program akan mencetak "Too Small".

**Langkah-6.**

Bagian 1: Simbol \* di bagian paling atas Mencetak simbol \* di bagian paling atas, dengan spasi yang sudah diatur oleh spasi0.

Bagian 2: Roti lapisan pertama(baris atas) Mencetak 3 baris berturut-turut dari simbol |-| (roti lapisan pertama), dengan spasi yang diatur oleh spasi1.

Bagian 3: Pemisah antara lapisan pertama dan kedua Mencetak garis pemisah menggunakan simbol -, di mana panjang garis sesuai dengan kolom2+2, dengan spasi yang diatur oleh spasi2.

Bagian 4: Roti lapisan kedua(bagian tengah) Mencetak baris tengah menggunakan simbol | dan =, di mana jumlah = menyesuaikan dengan nilai kolom2, dan dicetak sebanyak baris2 kali.

Bagian 5: Pemisah antara lapisan kedua dan ketiga Mencetak garis pemisah antara lapisan kedua dan ketiga menggunakan simbol -, dengan panjang sesuai kolom3+2.

Bagian 6: "Roti lapisan ketiga" (bagian bawah) Mencetak lapisan ketiga di bagian bawah menggunakan simbol | dan =, dengan jumlah = sebanyak kolom3, dicetak sebanyak baris3 kali.

Bagian 7: Pemisah terakhir di bagian bawah Mencetak garis pemisah terakhir menggunakan simbol -, dengan panjang sesuai kolom3+2.